



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**MOBILNÍ APLIKACE PRO POŘIZOVÁNÍ NOVÝCH
ZÁBĚRŮ STARÝCH POHLEDNIC**

MOBILE APP FOR SHOOTING NEW PHOTOS OF ANCIENT POSTCARDS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN BUCHTA

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, Ph.D.

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Buchta Martin**

Obor: Informační technologie

Téma: **Mobilní aplikace pro pořizování nových záběrů starých pohlednic**
Mobile App for Shooting New Photos of Ancient Postcards

Kategorie: Uživatelská rozhraní

Pokyny:

1. Seznamte se s problematikou "re-photography", vyhledejte a popište existující nástroje a přístupy.
2. Navrhněte uživatelské rozhraní pro pořizování nových záběrů starých snímků - zaměřte se na použitelnost širokým spektrem uživatelů a jednoduchost.
3. Prototypujte a vyvíjejte navržené rozhraní, experimentujte s různými variantami dílčích řešení.
4. Na základě provedených experimentů navrhněte a vytvořte aplikaci pro Android, připravenou pro reálný provoz mezi laickými uživateli.
5. Testujte aplikaci na dostatečném počtu uživatelů. Vyhodnoťte a vylepšujte vytvořené uživatelské rozhraní.
6. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 a 4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Herout Adam, prof. Ing., Ph.D., UPGM FIT VUT**

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Bc Zetechova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cílem práce je vytvořit mobilní aplikaci umožňující pořizování aktuálních snímků historických pohlednic. Aplikace je implementována pro operační systém Android. V práci jsou popsány teoretické základy o problematice repeat photography a o použitých technologiích. V praktické části práce je popsán návrh, implementace a testování aplikace. Pro praktickou část bylo použito vývojové prostředí Xamarin, které je založeno na programovacím jazyku C#. Výsledná aplikace umožňuje uživateli pořídit co nejpřesnější snímek odpovídající historické pohlednici.

Abstract

The aim of this thesis is to create a mobile application which allows to capture actual photographs of ancient postcards. Application is implemented for Android operation system. Theoretical foundations about problematics of repeat photography and used technologies are described in this thesis. Practical part of this thesis describes design, implementation and testing of the application. Xamarin integrated development environment which is based on programming language C# was utilized for practical part of thesis. Final application allows user to capture the most accurate photography corresponding to historical postcard.

Klíčová slova

staré pohlednice, historická fotografie, repeat photography, re-photo, mobilní aplikace, Android, Xamarin

Keywords

ancient postcards, ancient photography, repeat photography, re-photo, mobile app, Android, Xamarin

Citace

BUCHTA, Martin. *Mobilní aplikace pro pořizování nových záběrů starých pohlednic*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Herout Adam.

Mobilní aplikace pro pořizování nových záběrů starých pohlednic

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana profesora Adama Herouta. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Martin Buchta

16. května 2017

Poděkování

Děkuji prof. Ing. Adamu Heroutovi, Ph.D. za odborné vedení bakalářské práce a cenné rady při jejím vypracování. Dále bych chtěl poděkovat svému kolegovi Martinu Štainerovi za vytvoření webové aplikace a Ing. Ivo Petrovi za poskytnutí databáze s historickými fotografiemi. Děkuji i své rodině a přítelkyni za podporu při psaní práce.

Obsah

1	Úvod	2
2	Motivace k vytvoření řešené mobilní aplikace	3
2.1	Řešení problematiky při repeat photography	3
2.2	Shromáždění historických pohlednic	4
2.3	Průzkum existujících řešení	4
2.3.1	Aplikace RePhoto	5
2.3.2	Aplikace WhatWasThere	5
3	Vývoj mobilních aplikací s pomocí frameworku Xamarin	8
3.1	Popis platformy Android	8
3.2	Xamarin.Forms	9
3.3	Google API	12
3.4	Komunikace se serverem	12
4	Návrh uživatelského rozhraní a implementace aplikace	13
4.1	Uživatelé a případy užití	13
4.2	Požadavky na aplikaci	14
4.3	Vývoj aplikace pomocí prototypování	14
4.4	Popis fungování aplikace	15
4.5	Použité prvky uživatelského rozhraní	16
4.6	Programová část vývoje aplikace	18
4.7	Vlastní implementace kamery	19
4.8	Propojení s webovou aplikací	20
4.9	Implementace uživatelského rozhraní	21
4.10	Problémy spojené s implementací mobilní aplikace	22
5	Testování	24
5.1	Aplikace verze 1.0	24
5.2	Aplikace verze 1.1	26
5.3	Finální verze aplikace	26
6	Závěr	28
	Literatura	29
	Přílohy	30
A	Obsah CD	31

Kapitola 1

Úvod

Tato práce popisuje tvorbu mobilní aplikace RePhoto pro operační systém Android. Aplikace slouží k pořizování nových záběrů historických pohlednic. Důvodem pro vypracování této práce je nedostatek existujících řešení: žádná z aplikací nemá propojení mezi mobilní a webovou aplikací. Projekt bakalářské práce byl původně plánován na platformu Android, jelikož podle statistik je to nejrozšířenější operační systém běžící na mobilních telefonech. Aplikace je vytvořena s použitím nástroje Xamarin a může být v budoucnu rozšířena i na další mobilní operační systémy.

Cílem aplikace je nabídnout lidem možnost zdokumentovat nové podoby míst, která byla vyfocena s časovou mezerou a zobrazit si jejich porovnání. Uživatel si pomocí aplikace zobrazí fotografie ve své blízkosti. Vybere si místo, které ho zaujalo a jakmile se zorientuje v terénu a najde přibližnou pozici, ze které byla fotografie pořízena, udělá aktuální záběr místa. Uživateli, kterému není známa přesná poloha místa, dopomůže k tomuto bodu navigace. Při pořizování nové fotografie je uživateli poskytnuta na pozadí historická podoba objektu. Posuvníkem si může měnit intenzitu jejího prolnutí. Pokud se mu nepovede vyfotit přesnou podobu místa, má možnost si ji pomocí gest na obrazovce zvětšit, nebo zmenšit a posunout tak, aby ji zasadil do kontextu historické fotografie. Uživatel má možnost přidat k nově pořízené fotografii komentář a odeslat ji na webovou aplikaci. Ve webové aplikaci si bude moci prohlížet veškeré vyfocené fotografie a jejich srovnání.

Tato práce byla vyvíjena jako start-up projekt a bude úspěšná, pokud zaujme co největší množství uživatelů. Aplikace tohoto typu je atraktivní pouze v případě, jestli bude mít uživatel možnost vyfotit nový záběr místa nejen v jednom městě, ale nejlépe kdekoli v České republice. Zajistili jsme databázi s celkovým počtem historických fotografií čítajícím přes 50 000 snímků.

V následujících kapitolách budu řešit motivaci, proč jsem aplikaci tvořil. Budu se zabývat problémy se současným vytvářením mobilních aplikací, a také možnostmi, jak je vyvíjet. Zmíním se i o samotné platformě Android a dalších technologiích použitých při vývoji. Dalším tématem bude vlastní návrh aplikace a způsob, jakým byla vyvíjena a popis použitých prvků uživatelského rozhraní. Dále popíši způsob implementace a veškerou komunikaci s webovou aplikací. Následující kapitola bude pojednávat o způsobu testování a předchozích verzích aplikace, jejich nevýhodách a plánech na vylepšení.

Kapitola 2

Motivace k vytvoření řešené mobilní aplikace

V současné době se lidé zajímají o místa, kde žijí a která si pamatují z dřívějších dob. Pozorují jejich úpadek, nebo na druhou stranu růst nových objektů. V časopisech se objevují články, které srovnávají podoby míst v různých dobách. Na Facebooku existují skupiny, které se zabývají historií měst a k historickým fotografiím přikládají aktuální podoby. Této skutečnosti pomůže projekt RePhoto, který se skládá z mobilní a webové aplikace, kterou řešil kolega Martin Štainer. Aplikace jsou vzájemně propojené, a proto dodají uživatelům více možností než jakékoliv existující alternativy.

Dalším krokem byl průzkum trhu. Byl jsem překvapen, že je velký nedostatek podobných aplikací. Našel jsem podobné projekty, jejichž funkčnost a rozdíly jsou popsány níže.

2.1 Řešení problematiky při repeat photography

Repeat photography je technika pořizování snímků míst, která byla už dříve vyfotografována. Úspěšnému vyfotografování stejného místa lze zjednodušeně dojít těmito kroky: najít místo, odkud byl historický snímek pořízen, správně nastavit kameru a pořídit nový záběr. K nalezení místa, odkud byla fotografie pořízena je velice složitý proces, na který se používají speciální techniky odhadu vzdálenosti od fotografovaného snímku. V některých případech je nemožné se na tyto místa dostat, kvůli nepřístupnému terénu. V současnosti se nejčastěji k nalezení místa používá informační systém GIS¹.

Repeat photography se využívá nejčastěji v oboru sledování klimatických změn (např. na obrázku 2.1 lze vidět snímky z Národní Rezervace Wolong²), avšak používá se i pro pozorování kulturních, zdravotních nebo osobních změn. Při sledování proměn krajiny je cílem zdokumentovat vliv přírodních změn. Tímto způsobem se mohou například testovat hypotézy o vlivu globálního oteplování na změnu krajiny nebo ústupu ledovců. Z hlediska zdravotních věd se používala repeat photography pro snímky změn na lidech, kteří trpěli nemocí. Byla potřeba pozorovat změny a vývoj nemocí, při pokusech o jejich léčbu. Používala se například při pozorování léčby tuberkulózy nebo při poškození sítnice [1].

Vzhledem k faktu, že tato bakalářská práce se zabývá opakovaným fotografováním historických pohlednic, je důležité rozebrat repeat photography z kulturního hlediska. Problémy spojené s pořizováním nových záběrů historických pohlednic jsou zejména překážky mezi

¹Geographical Information System – <http://www.esri.com/what-is-gis>

²Převzato z: <http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0025008>



Obrázek 2.1: Na obrázku je vidět ukázka repeat photography z hlediska sledování krajiny. Na obrázcích je vidět snímek z Národní Rezervace Wolong v Číně.

budovami. Poněvadž se města neustále rozšiřují a staví se nové budovy, často není možné najít původní místo, ze kterého byl pořízen historický snímek. Dalším problémem je fotografování pohlednic z vyšších pater budov. Ukázka úspěšného pořízení nové fotografie stejného místa je na obrázku 2.2.



Obrázek 2.2: Na následujících obrázcích je zobrazeno porovnání podoby města v roce 1960 a 2017.

2.2 Shromáždění historických pohlednic

Pro správné fungování aplikace byla potřeba zajistit větší množství historických pohlednic, poněvadž při malém množství by práce nedokázala zaujmout větší spektrum uživatelů. Tento problém byl řešen před rozhodnutím, zda začít projekt realizovat. Navštívili jsme několik archivů města Brna a města Olomouce. Tam nám bylo většinou sděleno, že digitální podoba historických fotografií míst není k dispozici nebo bychom si snímky museli koupit. Poté jsme kontaktovali pana Ing. Ivo Petra, který vlastní webové stránky Fotohistorie [6] a byl ochotný s námi komunikovat. Při osobní schůzce nám poskytl oprávnění využít jeho databázi fotografií, která obsahuje přes 50 000 fotografií zajímavých míst po celé České republice.

2.3 Průzkum existujících řešení

Při průzkumu existujících řešení jsem narazil na několik mobilních aplikací, které se k historickým fotografiím vztahují. Jejich funkčnost je v mnoha ohledech různá od řešené bakalářské práce. Žádná z nalezených aplikací není v českém jazyce, tudíž není ani dostupná

dostatečná databáze fotografií na území České republiky. Všechny srovnávané aplikace byly staženy zadarmo na oficiálním obchodě s aplikacemi Google Play³.

2.3.1 Aplikace RePhoto

Aplikace RePhoto⁴ je v oblasti znovupořizování historických fotografií nejbližší řešenému projektu. Implementace je dostupná pro iOS i pro Android. Vzhledem k dostupným prostředkům byla otestována pouze na zařízení s operačním systémem Android. Při spuštění aplikace se otevře stránka s mapou, která je vycentrována na nejbližší místo pořízení historické fotografie. Při klepnutí na bod se otevře stránka, která umožňuje focení aktuálního snímku prolnutého historickým záběrem místa. Je možné si zvolit intenzitu prolnutí fotografií a také si přepnout, jak velkou část obrazovky má prolnutí zabrat. Při pořízení záběru je možnost prolnutí fotografií a uložení nově pořízeného snímku, nebo zrušení celé akce. Při uložení se fotografie nahraje na webovou aplikaci projektu. Názorná ukázka aplikace se nachází na obrázku 2.3.

Srovnání aplikací

Srovnávaná aplikace je založena jako projekt pro konkrétní uživatele, kteří si založí svou vlastní pozorovací oblast, kterou si v průběhu času znovu fotí. Existující oblasti se příliš nezajímají o historické fotografie, ale o pořizování snímků z nedávné doby. Příkladem jsou záběry okolí Washingtonské Univerzity, které se zajímají především o monitorování růstu stromů. Další z funkcí aplikace je pořizování záběrů pro osobní účely, kde si mohou lidé například zobrazit postup při rekonstrukci bytu.

2.3.2 Aplikace WhatWasThere

Další aplikace, která se zabývá problematikou znovupořizování fotografií se nazývá WhatWasThere⁵. Jedná se o webovou a mobilní aplikaci. Její provedení je řešeno způsobem nahrávání historických fotografií a zasazení do kontextu pomocí technologie Street View⁶ od firmy Google.

Pro přidání snímku se uživatel musí přihlásit pomocí e-mailu a hesla. Nahraje vybranou fotografii a přidá k ní titulek, popis a rok pořízení. Pokud uživatel nezná rok pořízení, aplikace nabídne volbu století, nebo lze nahrát fotografii i bez tohoto údaje. Po vyplnění má uživatel možnost vybrat si pozici pořízení snímku a zasadit ho do finálního kontextu. K tomu mu dopomohou nástroje, jako je zmenšení nebo zvětšení původního záběru. Po uložení snímku si ho může prohlédnout ve svém albu nahraných fotografií.

Srovnání aplikací

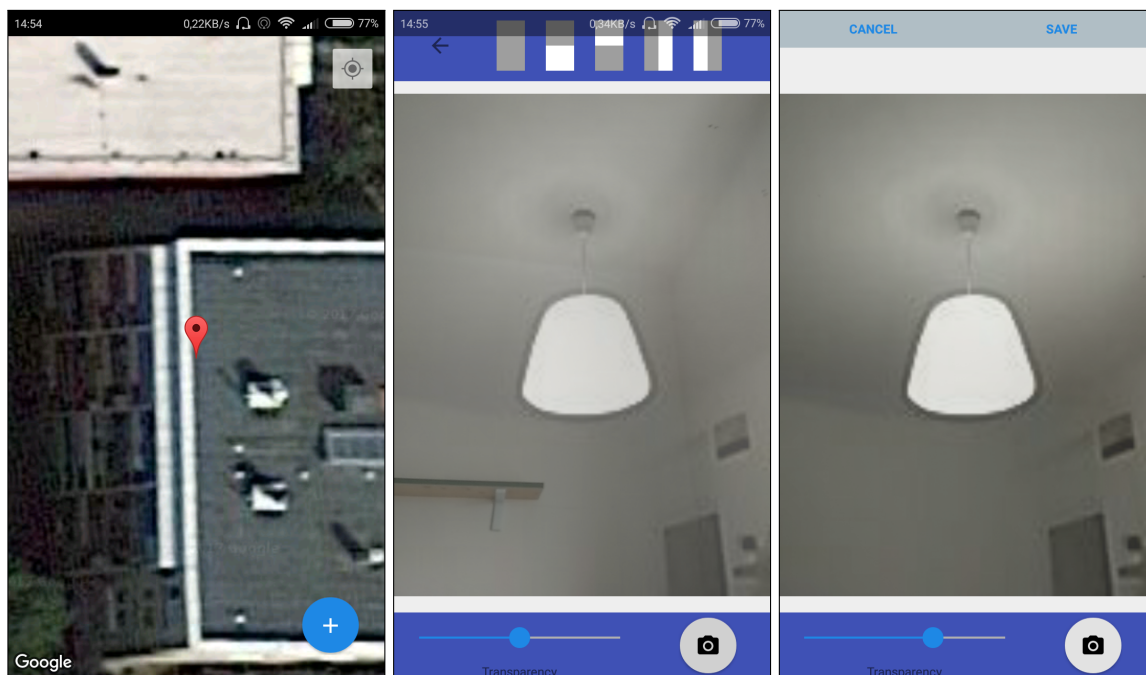
Mobilní aplikace existuje pouze pro platformu iOS, pro jejíž testování jsem neměl dostupné prostředky. Při zkoumání webové aplikace jsem narazil na problém u samotného řešení, jak zasadit fotografii do kontextu. Snímky pořízené přes Street View neumožňují uživatelům nahrát fotografie, které byly pořízeny ve větší vzdálenosti od silnice. Další problém představuje přesná poloha záběru. Není možnost změřit vzdálenost mezi dvěma snímky, protože je

³Dostupné z: <https://play.google.com/store>

⁴Dostupná z: <https://play.google.com/store/apps/details?id=com.cscience>

⁵Dostupná z: <http://www.whatwasthere.com/>

⁶Dostupné na: <https://www.google.cz/intl/cs/streetview/>



Obrázek 2.3: Na následujících třech obrázcích je zobrazena ukázka z konkurenční aplikace RePhoto. Na obrázku vlevo lze vidět úvodní stránku při spuštění aplikace. Na prostředním obrázku je vidět aplikace ve stádiu focení. Na obrázku vpravo lze vidět rekapitulaci pořízeného snímku s jeho vzorem. Je zde možnost uložení nebo zrušení.

závislá na rychlosti a GPS signálu vozu. Ve městě je vzdálenost přibližně 10 metrů, mimo něj se interval zvyšuje. I délka deseti metrů může vytvořit uživateli potíže s určením přesné pozice fotografie. Neméně podstatnou nevýhodou jsou překážky mezi silnicí a objektem (např. stromy, ploty). Tento druh zábrany je zobrazen na obrázku 2.4. Bakalářská práce tyto problémy řešit nemusí, protože dodává uživateli větší interakci s pořizováním snímků.

Ostatní aplikace

Výše zmíněné aplikace byly nejeefektivnější v problematice znovupořizování fotografií. K jedné z dalších zkoušených aplikací patřila Rephotos⁷, která při testování neplnila funkčnost popsanou na oficiálních stránkách Google Play. Aplikace má nahrávat fotografie z paměti telefonu a umožnit uživateli jejich znovupořizování. Při pokusu o nahrání fotografie aplikace zobrazila chybu. Stránka, která umožňovala focení, dokázala přepínat mezi přední a hlavní kamerou, ale obraz byl sekaný a v podstatě nepoužitelný.

Další z aplikací je HistoryPin⁸, která má velice podobnou logiku přidávání fotografií jako WhatWasThere, a proto ji nebudu příliš detailně rozebírat. Při zkoušce aplikace jsem zjistil, že není přímo zaměřena na zasazení snímku do kontextu historického záběru. Uživatel historickou fotografii může připnout pouze na místo, kde se zhruba nachází. Tato možnost je rozšířena o její přesné připnutí pomocí Street View.

⁷Dostupná na: <https://play.google.com/store/apps/details?id=alcantech.RePhotos&hl=cs>

⁸Dostupná z <https://www.historypin.org>



Obrázek 2.4: Na obrázku je vidět ukázka z webové aplikace WhatWasThere, která se zabývá příkládání fotografií pořízených z automobilu Google Street View. Na příkladu lze vidět nesrovnalost při zasazení fotografie do kontextu, protože se v pohledu nachází překážka.

Kapitola 3

Vývoj mobilních aplikací s pomocí frameworku Xamarin

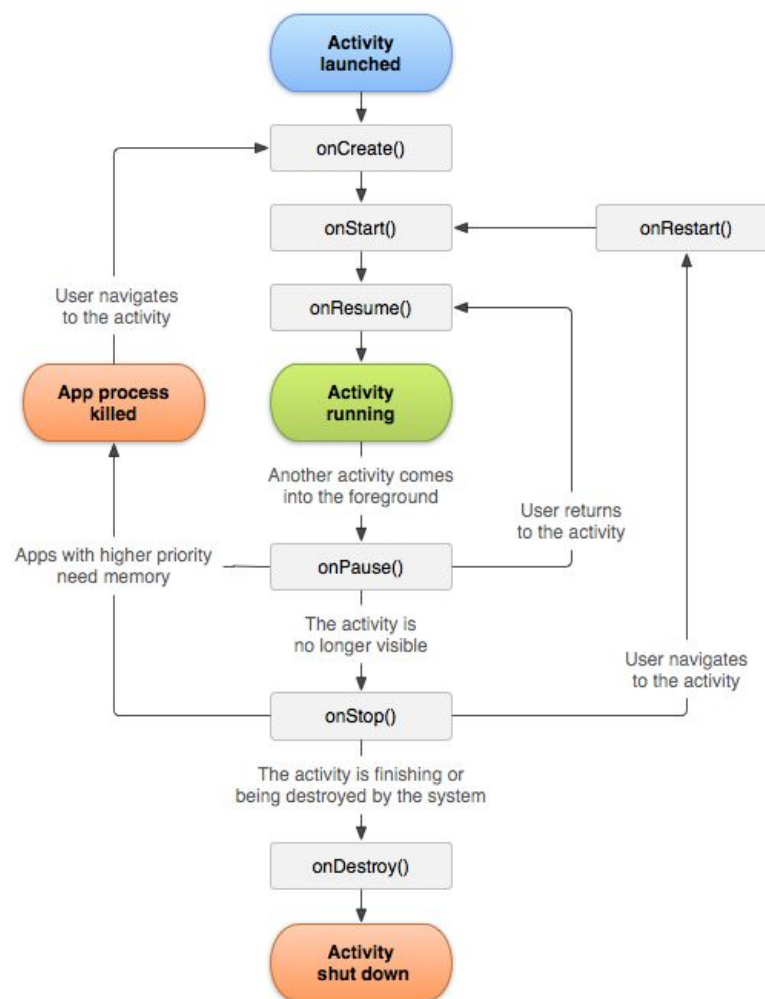
Xamarin je softwarová firma, která byla založena v roce 2011. Vznikla z firmy Ximian, která vytvořila projekt Mono. Vyvíjí sadu nástrojů pro multiplatformní vývoj zvanou Xamarin. Projekt Mono byl vyvinut společností Novell a Xamarin byl založen, aby ve správě projektu pokračoval. Mono je open source implementace rozhraní .NET, kterou lze použít k vývoji kompatibilních aplikací na více platformách. Do doby vzniku frameworku byl vývoj mobilních aplikací nejrozšířenější v programovacím jazyku Java a vývojovém prostředí Android Studio, který je dodnes nejčastěji volenou variantou pro tvorbu mobilních aplikací.

3.1 Popis platformy Android

Android je operační systém založený na jádře Linuxu. Jeho vývoj je pod záštitou společnosti Google. Je to open source software. Je to nejrozšířenější mobilní operační systém následovaný Apple iOS a Microsoft Windows Phone (viz. tabulka 3.1). Jeho uplatnění lze nalézt nejen u mobilních zařízení a tabletů, ale má širokou škálu využití (např. Android TV, Android Wear atd.). Vývoj aplikací na Android je bezplatný. Další věcí je zveřejnění aplikací přes Google Play, které je zpoplatněno jednorázovou částkou 25 USD. Po zaplacení je zpřístupněné nejen zveřejnění aplikace, ale i jiné výhody od Google jako je Google API atd. Další možností je zveřejnění bez poplatků na Amazon Store, který ale není tak rozšířený jako Google Play. Každý programátor musí počítat s tím, že je větší množství verzí Android platformy a ne vždy je aplikace kompatibilní se všemi zařízeními. Pro aplikace, které by měly zaujmout co nejvíce uživatelů, by programátor měl vybírat implementační prvky, které jsou kompatibilní pro co největší rozmezí verzí.

Jedním ze základních komponent Android je **Activity**. Aplikace obvykle sestává z více aktivit. Většinou se po startu aplikace spustí hlavní aktivita, která je definovaná v souboru **AndroidManifest.xml**. Aktivita běžně spouští nové okno, které zaplní celou obrazovku a pomocí **Layoutu** zobrazuje prvky na stránce. Zobrazení nebo provedení aktivity má často vysokou náročnost, proto jejich chod řídí **Activity Manager**. Každá z nich má svůj životní cyklus, který je zobrazený na obrázku 3.1.

²Převzato z: <https://developer.android.com/reference/android/app/Activity.html>



Obrázek 3.1: Na obrázku lze vidět životní cyklus aktivity. Ta je základním prvkem při tvorbě aplikací pro operační systém Android.²

3.2 Xamarin.Forms

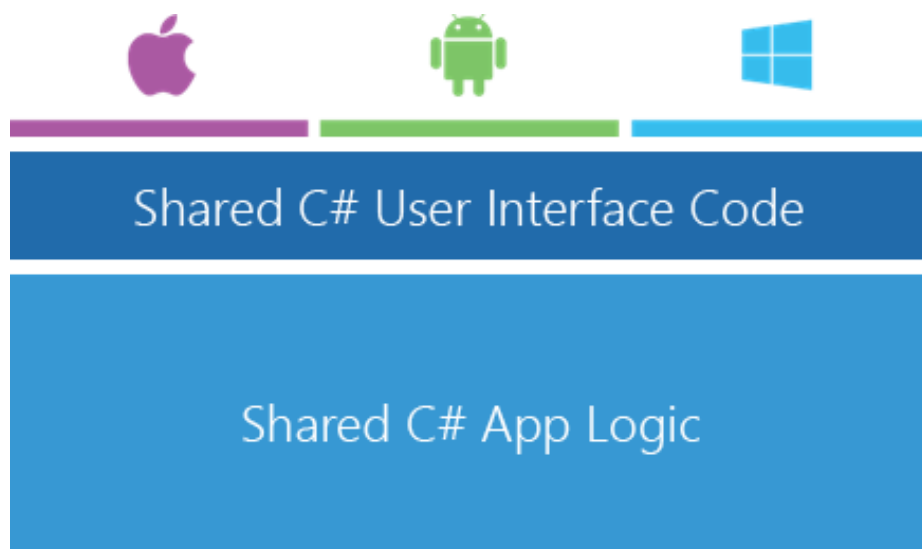
Xamarin Forms je multiplatformní UI (User Interface – uživatelské rozhraní), které umožňuje vývojářům snáze vytvářet uživatelské rozhraní společné pro Android, iOS a Windows Phone. Schéma vývoje v Xamarin.Forms je popsáno na obrázku 3.2. Prvky, které jsou psány ve sdíleném kódu, lze využít pro jednoduchý design a běžné ovládání. Avšak pro vývoj složitějších projektů, kde jsou potřeba animace, složitější ovládání prvků, nativní mobilní dialogová okna nebo vlastní přizpůsobení mapy, je potřeba renderování. Pro každou platformu musí programátor vytvořit vlastní renderer, ale všechny jsou nad společným kódem. Tyto renderery umožňují programátorovi využít specifitějších vlastností prvků uživatelského rozhraní každé platformy [5].

Tvorba mobilní aplikace pro každou platformu zvláště skýtá několik problémů [7]:

- **Různá paradigmatu uživatelského rozhraní** – všechny platformy mají podobné cesty, jak zobrazit grafické prvky uživatelského rozhraní a podobné způsoby jejich

Operační systém	Podíl na trhu [%]
Android	81,7
iOS	17,9
Windows Phone	0,3
Ostatní	0,1

Tabulka 3.1: V tabulce lze vidět podíl mobilních operačních systémů na trhu [8]. Tato statistika byla vytvořena ve čtvrtém čtvrtletí roku 2016.



Obrázek 3.2: Schéma multiplatformního vývoje aplikace pomocí Xamarin.Forms. Ze schématu lze vyčíst, že všechny platformy jsou psány pod jedním sdíleným kódem uživatelského rozhraní a celá logika fungování je oddělena od této vrstvy.

ovládání, avšak způsob implementace je rozdílný. Každá platforma má jiný styl navigace mezi stránkami, prezentaci grafických prvků a jejich použití, dokonce i jiné zpracování dotyků fyzického zařízení. Každá platforma má svou vlastní kulturu vývoje a tyto kulturní konvence ovlivňují programátory v mnoha směrech.

- **Jiná vývojová prostředí** – programátoři jsou zvyklí pracovat ve svém oblíbeném vývojovém prostředí (IDE³). Pro každou z platforem takové IDE existuje, ale jsou v mnoha směrech velice rozdílná.

- Pro iOS se vyvíjí v Xcode.
- Pro Android existuje Android Studio.
- Pro vývoj Windows aplikací se používá Visual Studio.

- **Různá programovací rozhraní** - všechny tři platformy jsou založené na jiném operačním systému s různými API. V mnoha případech se implementuje stejný typ objektů uživatelského rozhraní, ale pod jinými jmény. Spousta z nich se neliší pouze

³Integrated development environment

ve jménu, ale i v jiných ohledech. Jako příklad je uveden objekt, který mění hodnotu `bool`⁴ operátoru. Například:

- Pro iPhone existuje „View“ se jménem `UISwitch`.
 - Na zařízeních se systémem Android je to „Widget“ se jménem `Switch`.
 - Ve Windows Runtime API je to „Control“ s názvem `ToggleSwitch`.
- **Jiný programovací jazyk** – programátor má omezenou možnost výběru programovacího jazyka pro každou z platforem. Obecně je přesto každá z nich úzce spjata s jedním programovacím jazykem. Všechny mají společný fakt, že jsou to objektově orientované jazyky.
 - Pro iPhone je to Objective-C.
 - Java pro Android zařízení.
 - C# pro zařízení s Windows.

Tyto problémy jsou velkým důvodem proč vytvářet mobilní aplikace multiplatformně.

Popis architektury MVVM

MVVM⁵ architektura je návrhový vzor. Jeho cílem je oddělit logickou stránku aplikace od jeho uživatelského rozhraní a dat. Tím se kód stává přehlednější a lépe strukturovaný. Název MVVM se skládá z prvních písmen tří slov, a to model, view a viewmodel [2]. Schéma MVVM je zobrazeno na obrázku 3.3.

- **Model** – popisuje data získaná z úložiště (např. databáze, webová služba). Model neví o stavu ovládacích prvků.
- **View** – v základě reprezentuje aktuální obrazovku, ve které se aplikace nachází i se všemi jeho ovládacími prvky (např. button, label atd.). Reprezentace je v jazyce XAML⁶.
- **ViewModel** – ovládá a mění views jako jejich datový kontext. Zprostředkovává transportní vrstvu mezi modelem a view. Drží stav aplikace.

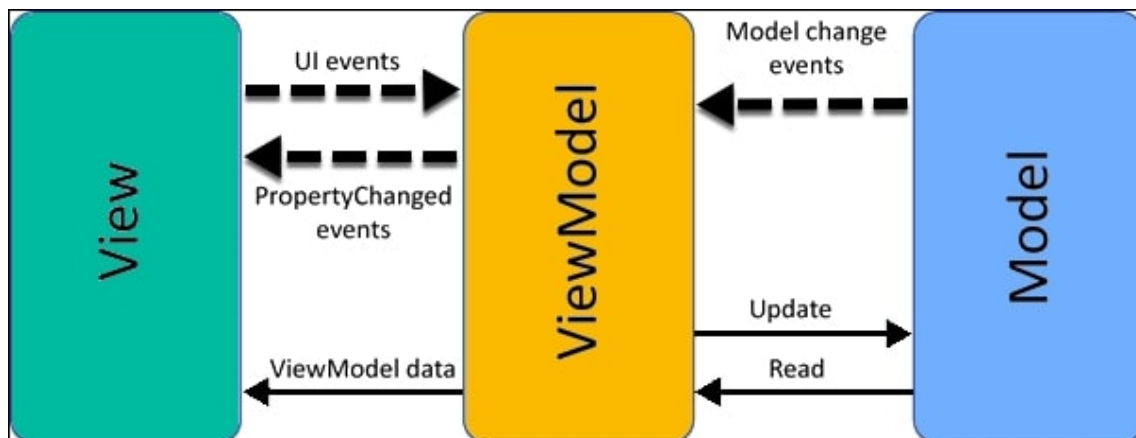
NuGet galerie

NuGet je open-source manažer balíčků použitelných pro vývojové platformy společnosti Microsoft. Celkový počet balíčků je několik desítek tisíc. NuGet galerie je centrální repozitář, který využívají všichni autoři balíčků i jejich spotřebitelé. Jakmile uživatel vybere balíček, který chce nainstalovat, NuGet manažer ho zkopíruje do jeho řešení a aktualizuje celý projekt včetně přiřazení referencí a změny v potřebných souborech.

⁴Boolean data type – https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.sqlr.doc/ids_sqr_099.htm

⁵Model-View-ViewModel – <https://msdn.microsoft.com/en-us/library/hh848246.aspx>

⁶Extensible Application Markup Language



Obrázek 3.3: Na obrázku je zobrazeno schéma Model-View-ModelView architektury.

3.3 Google API

Nedílnou součástí aplikace je API od Google, která poskytuje množství rozhraní pro programování aplikací. Google API byla vytvořena firmou Google a umožňuje programátorům komunikaci s Google službami a její integraci s nimi a ostatními službami. Mezi nejpoužívanější patří Search, Google Maps, Gmail, Translate a Youtube. Pro vytvoření aplikace jsem využil Google Maps k zobrazení míst, kde se nachází historické fotografie.

Google Maps

V aplikaci je využívána Google Maps API, kterou vyvíjí společnost Google. Existují čtyři základní typy zobrazení mapy: satelitní, hybridní, mapa zobrazující terén a mapa silnic. Pro aplikaci byla nejvhodnější právě mapa silnic, na které jsou zvýrazněné silnice a města. Pro zprovoznění Google Maps API je nutné vygenerovat klíč. Na tento klíč se poté musí mobilní aplikace odkazovat. Toto opatření je nutné kvůli počtu zobrazení mapy, které je omezené pro uživatele, kteří tuto API používají bezplatně.

3.4 Komunikace se serverem

Pro komunikaci se serverem jsou využívány metody HTTP⁷ protokolu. V aplikaci jsou používány pouze 2 metody:

- GET – je metoda pro odeslání požadavku na objekt, se kterým se odesílají data potřebná k jeho získání.
- POST – metoda k odesílání většího toku dat.

Způsob zápisu, který byl pro komunikaci použitý, se nazývá JSON⁸. Je to formát pro výměnu dat, který je založen na podmnožině programovacího jazyka JavaScript. Je založen na dvou strukturách. První z nich struktura párů název–hodnota a druhou je seřazený seznam hodnot [3].

⁷Hypertext Transfer Protocol – <https://www.computerhope.com/jargon/h/http.htm>

⁸JavaScript Object Notation – <https://www.thoughtco.com/what-is-javascript-2037921>

Kapitola 4

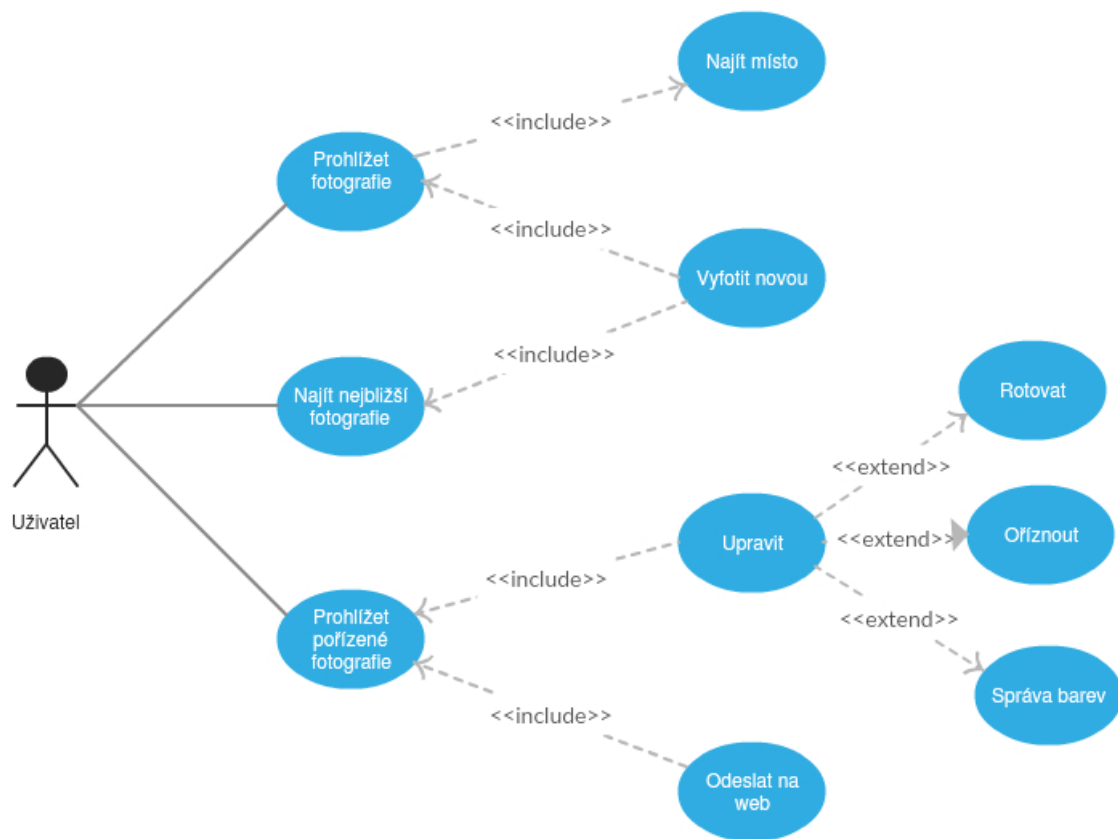
Návrh uživatelského rozhraní a implementace aplikace

Tato kapitola se bude zabývat požadavky na aplikaci, návrhem grafického rozhraní a datových modelů. Dále zde bude pojednáno o výběru cílové verze operačního systému a programovacím jazyku, ve kterém aplikace je vytvořena. Je zde rozveden i výběr vhodných implementačních prostředků.

4.1 Uživatelé a případy užití

Aplikace je zaměřena na všechny věkové skupiny. Na obrázku 4.1 je znázorněn diagram případů užití, který byl vytvořen před samotnou implementací a v mnohém byl upraven. Pro pochopení funkčnosti aplikace je potřeba si případy užití detailněji rozebrat:

- **Prohlížet fotografie** – prohlížení historických fotografií pomocí jejich výpisu ve formě snímku a informacemi o něm.
- **Najít místo, najít nejbližší fotografie** – jsou v konečné fázi spojeny do společného případu užití, kde jsou tyto body zobrazeny na mapě.
- **Vyfotit novou fotografii** – nejdůležitější funkce celé aplikace, pořízení aktuální fotografie místa, které je zobrazeno na historickém snímku.
- **Prohlížení pořízené fotografie** – po přezkoumání tohoto případu je lepší, když danou funkci bude vykonávat webová aplikace.
- **Upravit fotografii** – možnost ořezání a posun pořízené fotografie před jejím odesláním.
- **Odeslat fotografii** – odeslání fotografie s informacemi o pořízeném snímku, GPS souřadnicemi a informacích o uživateli na webovou aplikaci.



Obrázek 4.1: Diagram případů užití

4.2 Požadavky na aplikaci

Výsledná mobilní aplikace musí splňovat tyto požadavky:

- Být spustitelná na mobilních telefonech s minimální verzí operačního systému Android 4.0.4.
- Umožnit uživateli registraci a přihlášení s propojením s webovou aplikací.
- Grafické prvky musí mít přijatelný vzhled a funkčnost pro ovládání prstem na malých displejích.
- Umožnit uživateli vytvoření nového záběru historické fotografie a její následné odeslání na webovou aplikaci.

4.3 Vývoj aplikace pomocí prototypování

Celý vývoj probíhal pomocí prototypování. U prvního návrhu jsem vycházel z již vytvořených případů užití, které jsou vidět na obrázku 4.1. Po vytvoření prvního funkčního prototypu jsem vycházel z poznatků potenciálních uživatelů a svého vedoucího bakalářské práce a aplikaci podle nich přetvářel.

Technika vývoje prototypováním je postavena na vytváření neúplných verzí softwaru. Každý nový prototyp se musí zároveň prodiskutovat s potenciálními uživateli a jejich poznatky o softwaru zaznamenat do další verze. Vývoj pomocí této techniky se vyplatí použít u softwaru, který má zaujmout velké spektrum uživatelů. Při počátečním návrhu se většinou tvoří různé mockupy¹ a další návrhy, které předchází samotné implementaci [4]. Při prototypování se naimplementuje první verze programu, jejíž prvky se podle určitého počtu potenciálních uživatelů upraví, ponechají, nebo úplně odstraní. Jednou z důležitých částí je testování na více zařízeních, a proto byly verze rozesílány potenciálním uživatelům v podobě instalačních souborů aplikace. Tímto způsobem lze nejlépe eliminovat problémy s GUI².

4.4 Popis fungování aplikace

Celá funkčnost mobilní aplikace je zobrazena na videu, které je uloženo na webové stránce³. Při spuštění aplikace by měla první stránka uživatele zaujmout. Vzhledem k tomu, že nelze přeskočit přihlašovací stránku, musí právě ta být navržena tak, aby uživateli přiblížila funkci aplikace. Toho je dosaženo přidáním koláže historických fotografií na pozadí na obrázku 4.2 vlevo. Dále je na stránce ikona pro nápovědu k aplikaci, kde je vysvětleno základní ovládání aplikace a její funkčnost. Přihlašování do aplikace je možné dvojím způsobem. První z nich je vytvoření vlastního účtu (na obrázku 4.2 uprostřed). K vytvoření vlastního účtu musí uživatel zadat své jméno a příjmení, uživatelské jméno, e-mail a heslo. Poté se přes REST API vytvoří záznam v databázi webové aplikace. Dále se na přihlašovací stránce přihlásí pomocí e-mailu a svého hesla. Druhým způsobem přihlášení je registrace přes webovou aplikaci, ve které si uživatel vygeneruje QR kód⁴ k přihlášení, který bude mít platnost jeden den. Po vygenerování si uživatel přes zabudovanou čtečku QR kódů (na obrázku 4.2 vpravo) načte svůj ověřovací symbol a přes REST API proběhne jeho autorizace.

Po úspěšném přihlášení je uživateli poskytnuta stránka s mapou (obrázek 4.3 vlevo), která je vycentrovaná na Českou republiku a jsou na ní zobrazena všechna místa, kde se nachází historická fotografie. Lze vidět i aktuální polohu mobilního zařízení, na kterou je možné se přemístit pomocí tlačítka aktuální pozice. Při přepnutí lokální navigace umístěné ve vrchní části stránky na „Výpis“ se objeví výpis všech zobrazených fotografií viditelných na mapě (obrázek 4.3 uprostřed). Tento počet fotografií se mění dynamicky při přiblížení na určité místo (např. když uživatel nechá mapu vycentrovanou na celou Českou republiku, bude ve výpisu více fotografií, než když bude mít mapu přiblíženou na oblast Králova Pole). Na stránce s výpisem si poté uživatel zvolí fotografii, která ho zaujala a chce ji vyfotit. Snímek se mu objeví na nové stránce (obrázek 4.3 vpravo), kde má možnost si k místu zobrazit podrobnosti, pokud jsou k dispozici, nebo pomocí externí aplikace navigovat pomocí GPS⁵ souřadnic k danému místu. Dále se může vrátit na předchozí stránku s výpisem záběrů nebo zvolit, že právě tento snímek chce vyfotit.

Jakmile uživatel zvolí možnost pořídit fotografii, otevře se nová stránka (obrázek 4.4 vlevo nahoře), která mu umožní aktuální snímek pořídit. Důležité je, aby byl nový záběr co nejlépe zarovnaný s fotografií historickou. Tomu dopomůže prolnutí obou fotek. Zároveň je zde i možnost si měnit pomocí posuvníku intenzitu prolnutí. Jakmile je fotografie pořízena, objeví se stránka (obrázek 4.4 vpravo nahoře), která umožní uživateli úpravu

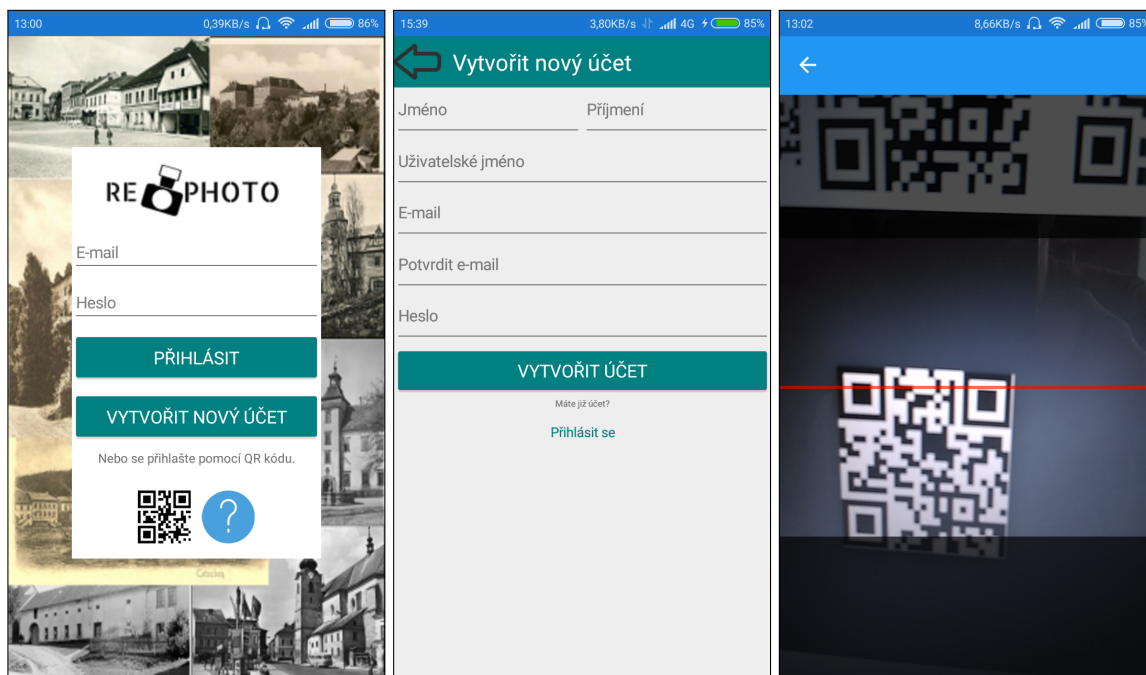
¹Dostupné na: <https://experience.sap.com/basics/what-is-a-mock-up/>

²Graphical User Interface – <https://www.computerhope.com/jargon/g/gui.htm>

³Dostupné na: <http://demo.egoba.cz/fotografie/video.mp4>

⁴Quick Response Code – <http://www.whatisaqr.com/>

⁵Global Positioning System – <http://www.gps.gov/>



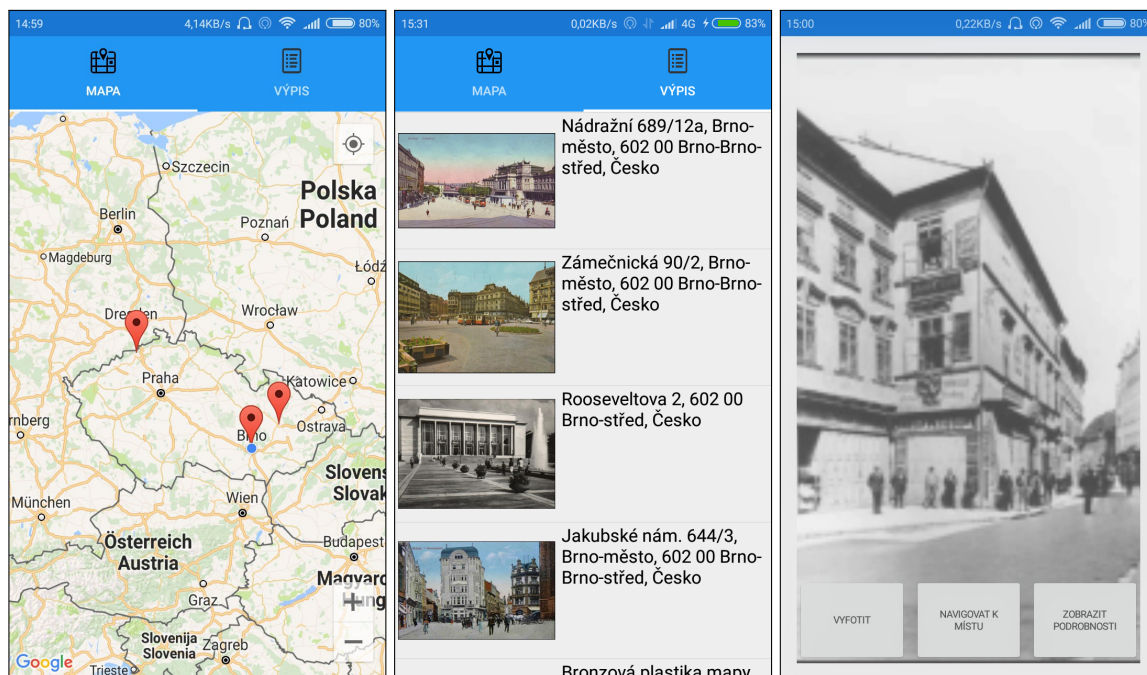
Obrázek 4.2: První ukázka z praktické části bakalářské práce – aplikace RePhoto. Na levém obrázku je vidět přihlašovací stránka, kde se uživatel pomocí přihlašovacích údajů do aplikace připojí. Na prostřední ukázce je vidět stránka s vytvořením vlastního účtu. Na obrázku vpravo je ukázka přihlašování přes QR kód, který je vygenerován webovou aplikací.

nově vytvořeného snímku. To je možné pomocí gest. První gesto je proveditelné pomocí dotyku dvou prstů na zobrazený snímek a jejich pohyb směrem od sebe nebo k sobě způsobí přiblížení a oddálení záběru. Druhé gesto se používá při přiblížené fotografii a způsobuje posun všemi směry. Dále se na této stránce nachází posuvník, který umožňuje stejnou funkci jako na předchozí stránce. Je zde i tlačítko náhledu, které umožňuje skrýt nebo zobrazit viditelnost historického snímku. Jakmile je uživatel spokojen s úpravou fotografie, klepne na tlačítko pro potvrzení fotografie a je přesunut na stránku (obrázek 4.4 uprostřed dole), kde má možnost přidat komentář k pořízenému záběru a odeslat na webovou aplikaci 4.5. Není-li fotografie vyfocena přesně a nelze už více upravit, aby byla zasazena do kontextu, klepne na tlačítko pro zrušení aktuálního snímku a je přesunut znovu na stránku umožňující focení.

4.5 Použité prvky uživatelského rozhraní

Jak už jsem se zmínil v sekci 3.2, ovládání je implementováno ve sdíleném kódu. Prvky uživatelského rozhraní jsou definovány v XAML souborech, které reprezentují vzhled stránek. Existuje více druhů stránek, ale popíšu ty, které jsou v aplikaci využity, a jejich umístění. Jejich vzhled je zobrazen na obrázku 4.6.

- **ContentPage** – základní jednoduchá stránka. Pro zobrazení prvků na této stránce se musí implementovat uvnitř této třídy. Je použita téměř pro všechny stránky, mimo stránky uvedené níže.
- **NavigationPage** – stránka, která ovládá navigaci na ostatní.



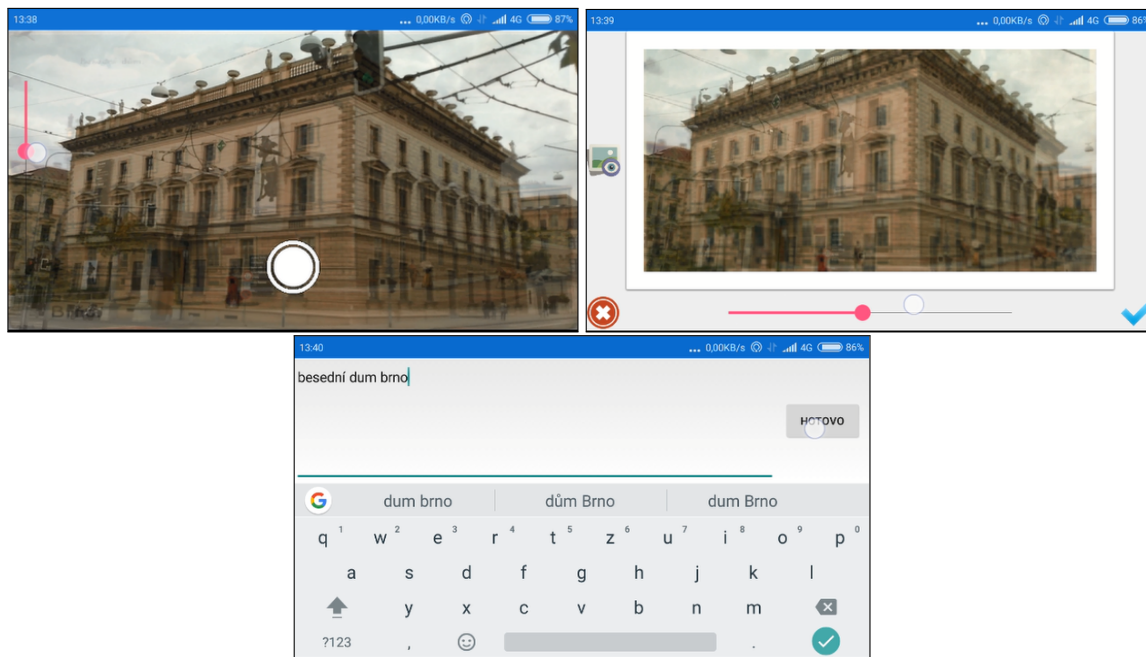
Obrázek 4.3: Na uvedených obrázcích pořízených z obrazovky při použití aplikace jsou zobrazeny ukázky stránek. Vlevo se nachází obrázek, který je složen z mapy a lokální navigace. Na mapě jsou zobrazeny body, které značí místo s historickou fotografií. Na prostředním obrázku je zobrazen výpis fotografií a informací o nich. Vpravo je zobrazena stránka, která umožní uživateli detailnější náhled fotografie a možnost navigace k ní, zobrazení poznámek nebo její vyfocení.

- **TabbedPage** – stránka složená ze dvou a více **ContentPage**, mezi kterými lze přecházet pomocí lokální navigace, která se nachází v horní části stránky. Přesun mezi stranami je možný i gesty tahu doprava nebo doleva. Lištu lze vlastním způsobem upravit a přidat text nebo ikonu.

Dále bylo využito několika druhů rozložení prvků po stránce. Je důležité zadávat hodnoty relativně, protože mobilní zařízení mají různé velikosti displejů. Nejvíce jsem používal rozložení pomocí **GridLayout**, které umožňuje nadefinovat si rozmezí řádků a sloupců a prvky do nich umisťovat. V mnoha případech je použito na jedné stránce více rozložení. Jedná se o kombinace těchto rozložení: **GridView**, **StackLayout**, **AbsoluteLayout**, **Frame** a **RelativeLayout**. Například v **GridView** používám pro zobrazení fotografie **StackLayout**.

Rozložení se používá pro rozmístění prvků na stránce. Některé z používaných prvků detailněji popíši:

- **Button** – tlačítko, které reaguje na klepnutí.
- **Label** – je to prvek, který zobrazuje text.
- **Image** – prvek, který zobrazuje obrázky. Lze nastavit jeho průhlednost, velikost nebo otočení. Je možné pro něj naimplementovat události, jako je klepnutí.
- **Entry** – prvek, jehož text lze editovat. Je možné nastavit přednastavené hodnoty nebo skrytí znaků při zadávání hesla.



Obrázek 4.4: Na uvedených obrázcích jsou zobrazeny stránky, které byly pořízeny z obrazovky telefonu při focení v terénu. Na obrázku nahoře vlevo je vidět stránka umožňující fotografování. Na levé straně je použit prvek **Slider**, který slouží ke změně průhlednosti historické fotografie. Fotografie je pořízena při klepnutí na další viditelný prvek **Button**. Na obrázku nahoře vlevo je ukázka ze stránky srovnání. Jsou zde umístěny tlačítka sloužící pro náhled historické fotografie (vlevo uprostřed), pro zrušení aktuálního snímku (vlevo dole) a pro potvrzení fotografie (vpravo dole). Stránka dole uprostřed slouží k přidávání komentářů k pořízenému snímku.

- **Map** – zobrazuje mapu a body na ní. Obsahuje množství gest, která slouží pro pohodlné ovládání mapy.
- **Slider** – posuvník, jehož hodnota se mění podle pozice bodu, se kterým je možné pohybovat.

4.6 Programová část vývoje aplikace

V této kapitole popíšu použité implementační prostředky, dále rozeberu hlavní problémy při implementaci a jejich řešení. Následně se budu zabírat propojením dat s webovou aplikací. Aplikace je psána v jazyce C# pro operační systém Android s minimální verzí 4.0.3. Většina kódu je přenositelná i na platformu iOS a Windows Phone. Uživatelské rozhraní je psáno pomocí Xamarin.Forms a je také přenositelné na ostatní mobilní operační systémy. Implementace na ostatní platformy neproběhla, a to kvůli celkové náročnosti. Dalším faktorem je nutnost aplikace pro iOS překládat na zařízení s OS X, které nemám k dispozici. První 3 měsíce jsem pro vývoj používal vývojové prostředí Visual Studio s pluginem Xamarin, které nebylo v mnoha ohledech vyhovující, a proto jsem začal používat IDE Xamarin Studio. Multiplatformní projekt pro Xamarin se skládá z několika částí. První z nich je sdílený kód pro všechny platformy. Další části jsou psány pro každou platformu zvlášť.



Obrázek 4.5: Na uvedeném obrázku je zobrazeno porovnání fotografií ve webové aplikaci.



Obrázek 4.6: Na obrázku vlevo je zobrazen typ stránky `ContentPage`. Prostřední obrázek se vztahuje k `NavigationPage`, která obstarává navigaci mezi stránkami. Na pravém obrázku lze vidět `TabbedPage`, která je složená ze dvou a více `ContentPage`.

4.7 Vlastní implementace kamery

Nastavení kamery a její ovládání nebylo možné implementovat ve sdíleném kódu, tudíž k němu musel být vytvořen renderer. Ve sdíleném kódu je naimplementován `EventHandler`, jenž řídí chod události, která má být provedena. Skládá se ze dvou částí. První částí je metoda, která zpracovává výsledek události. Druhou částí je třída, do které se ukládají data získané z výsledku události. Celkový vzhled stránky spočívá v rendereru. Jakmile

uživatel zvolí vyfocení historické fotografie, renderer spustí kameru. K jejímu spuštění musí uživatel při prvním použití aplikace povolit přístup ke kameře. Poté proběhne nastavení kamery a vykreslení prvků této stránky, mezi které patří `imageview`, ve kterém je zobrazena historická fotografie, `slider` pro změnu intenzity prolnutí `imageview` a `button`, jenž řídí samotné pořízení snímku. Kamera nastavuje svou orientaci podle podílu šířky a výšky historického snímku, který chce uživatel pořídít.

4.8 Propojení s webovou aplikací

Veškerá data jsou uložena na serveru, a proto je potřeba naimplementovat snadnou komunikaci s webovou aplikací. Aplikace počítá s připojením k internetu. Veškerá komunikace se serverem probíhá pomocí POST a GET metod protokolu HTTP, který už byl popsán v sekci 3.4. K přenosu dat mezi mobilní a webovou aplikací dochází v těchto případech:

- **Vytváření účtu** – mobilní aplikace odesílá registrační údaje metodou GET.
- **Přihlašování** – aplikace odesílá přihlašovací údaje nebo načtený qr kód metodou POST, v obou případech dochází ke kontrole odpovědi serveru. Pokud nenalezne uživatele v databázi vrací `false` a přístup je uživateli zamítnut. V opačném případě se uživatel přihlásí. Formát odesílaných dat je vidět v kódu 4.1.

```
1 public class Login
2 {
3     public string email { get; set; }
4     public string password { get; set; }
5 }
6 public class loginQR
7 {
8     public double QRCode { get; set; }
9 }
```

Zdrojový kód 4.1: Kód znázorňující třídu, jejíž objekt nese data, která jsou odesílána na webovou aplikaci při přihlašování.

- **Zobrazení pinů na mapě** – Po přihlášení okamžitě dojde k požadavku GET na určitou url, kde server vrátí GPS souřadnice všech míst, kde se nachází historická fotografie. Tyto body jsou pak zaznačeny do mapy.
- **Zobrazení fotografií ve výpisu** – Při každém pohybu na mapě a jejím zvětšení nebo zmenšení dochází k přepočítání souřadnic, které ohraničují viditelnou část mapy na obrazovce. Při přechodu na stránku s výpisem dojde k požadavku POST, kde se zároveň odešlou souřadnice pravého horního a levé dolního rohu mapy. Jako odpověď jsou vrácena všechna místa i s jejich údaji. Ty jsou detailněji zobrazeny v kódu 4.2.

```

1 public class PhotoModel
2 {
3     public string picture_url { get; set; }
4     public int id { get; set; }
5     public double width { get; set; }
6     public double height { get; set; }
7     public string comment { get; set; }
8     public double latitude { get; set; }
9     public double longitude { get; set; }
10    public string n_address { get; set; }
11 }

```

Zdrojový kód 4.2: Kód znázorňuje třídu, jejíž objekt nese data, které se přijímají z webové aplikace při odeslání souřadnic aktuálního rozložení mapy na obrazovce.

- **Odeslání aktuálního snímku na webovou aplikaci** – Před odesláním na webovou aplikaci se nejdříve fotografie zakóduje do base64⁶ a poté se i s dalšími potřebnými daty odesílá na webovou aplikaci. Odesílaná data jsou zobrazeny v kódu 4.3.

```

1 public class PhotoToSend
2 {
3     public byte[] image { get; set; }
4     public int historicalPicId { get; set; }
5     public int userId { get; set; }
6     public string comment { get; set; }
7     public double x { get; set; }
8     public double y { get; set; }
9     public double scale { get; set; }
10 }

```

Zdrojový kód 4.3: Kód znázorňující třídu, jejíž objekt nese data při odesílání aktuální fotografie na webovou aplikaci.

O celou funkčnost rozesílání požadavků a zpracovávání odpovědí se stará třída `RestService`. Všechna data, která se odesílají nebo přijímají, jsou nejdříve serializována nebo deserializována do formátu JSON, o kterém je zmínka v sekci 3.4.

4.9 Implementace uživatelského rozhraní

Jak už bylo zmíněno v sekci 3.2, tak uživatelské rozhraní se implementuje v jazyce XAML a je odděleno od logické části kódu. Jednotlivé prvky se propojují s logickou částí pomocí uvedení jména prvku.

Pro navigaci mezi stránkami je využita `NavigationPage`, kterou představuje přihlašovací stránka. Pro přesun na další stránku je využita metoda `PushModalAsync()`. Stránka, na kterou uživatel chce přejít, je načtena na zásobník a přesune se do popředí obrazovky. Pro odstranění stránky ze zásobníku je využita metoda `PopModalAsync()`, která se volá

⁶Dostupná na: <https://www.lifewire.com/base64-encoding-overview-1166412>

po klepnutí na tlačítko zpět nebo při klepnutí na tlačítko zpět na fyzickém zařízení. Rozdílné jsou stránky s mapou a výpisem, kde se využívá přechodu bez využití zásobníku.

Jednotlivé ovládací prvky jsou většinou rozmístěny na stránce pomocí `GridLayout`, u kterého si lze navolit výšky řádků a šířky sloupců a do uvedené tabulky přiřazovat prvky. Prvky uživatelského rozhraní jsou pozměňovány atributy. Každý prvek má různé druhy a omezené možnosti atributů. Nejčastěji se používají pro rozložení prvků, jejich umístění, velikost ale také pro změnu barvy nebo textu.

K ovládání prvků uživatelského rozhraní je potřeba implementovat funkce, které události řídí. Téměř každý prvek může mít implementovanou funkci pro událost. U `Button` jsou používány funkce `OnClick()`, pro `Image` je používána funkce `OnTap()` a u `ViewCell`, který představuje výpis fotografií je využívána funkce `OnSelection()`, která se vykoná, jakmile uživatel vybere obrázek.

Použité NuGet balíčky

K řešení aplikace byly použity některé z dostupných NuGet balíčků o kterých jsem se zmínil již v sekci 3.2. Použití balíčků vedlo ke zjednodušení celkové práce. Následuje výpis a popis funkčnosti konkrétních balíčků, které jsou v projektu obsaženy a využívány.

- **Xamarin.Forms** – balíček pro vývoj nativního UI pro iOS, Andorid a Windows Phone sdíleným kódem.
- **Xamarin.Android.Support.v4 a Xamarin.Android.Support.v7** – základní balíčky pro vývoj aplikací na platformu Android, obsahující reference pro provázání celého řešení.
- **Newtonsoft.Json** – balíček obsahující nástroje pro serializaci a deserializaci dat do formátu JSON.
- **Microsoft.Net.Http** – obsahuje HTTP klienta, který obstarává odesílání HTTP požadavků.
- **Xam.Plugin.Geolocator** – využít pro reverzní překlad GPS souřadnic na adresu.
- **Xamarin.Forms.Maps** – balíček pro práci s Google mapami. Obsahuje modely map a jejich renderery.
- **ZXing.Net.Mobile.Forms** – poskytuje scanování QR kódů potřebných pro přihlášení do aplikace.
- **Square.Picasso** – využito u načítání fotografií z URL.

4.10 Problémy spojené s implementací mobilní aplikace

V souvislosti s vývojem mobilní aplikace jsem se setkal s několika problémy, které v této sekci popíši. Problémy se týkaly vývojového prostředí, map od firmy Google a přihlašování do aplikace.

- **Problémy ve vývojovém prostředí** – první verzi aplikace jsem se rozhodl vyvíjet ve vývojovém prostředí Visual Studio s podporou nástroje Xamarin. Toto IDE často

hlásilo spoustu chyb v částech kódu, které žádnou chybu neobsahovaly. Při jejich řešení jsem strávil mnoho času hledáním příčiny těchto chyb. Většinou byly spojeny s chybným propojováním referencí a k jejich řešení se musel celý projekt vyčistit nebo restartovat celé vývojové prostředí a chyby se odstranily. Dalším problémem bylo přidávání NuGet balíčků. IDE často zakazovalo instalaci, protože kolidovaly s jiným balíčkem, který měl jinou funkčnost, nebo jsem pro jejich přidání musel odstranit všechny ostatní balíčky a poté je znovu nainstalovat a až po tomto kroku IDE tuto možnost povolilo. Při přechodu k vývojovému prostředí Xamarin Studio se tyto problémy neeliminovaly, ale chyb bylo hlášeno razantně méně.

- **Problémy s mapami** – NuGet balíček mapy od společnosti Google pro Xamarin.Forms implementuje pouze základní funkce pro práci s mapou. Tyto funkce často nebyly dostatečné. Jednou z nich je zobrazování detailů bodu na mapě, kde lze zobrazit pouze text. Další funkcí, která není možná, je sjednocování bodů na mapě do shluků při větším počtu fotografií na jednom místě. Aplikaci plánuji v budoucnu rozšířit o vlastní implementaci těchto funkcí.
- **Problémy s přihlašováním** – původní návrh se zabýval přihlašováním do aplikace přes vlastní účet, nebo přes účet na sociální síti Facebook nebo přes Google účet. Úspěšně bylo implementováno přihlašování přes sociální sítě použitím implementačního prvku `WebView`. Společnost Google na začátku roku 2017 oznámila konec podpory přihlašování přes tento prvek, tudíž byl tento přístup nefunkční a musel být odstraněn. Byly navrženy alternativy, ale nebyly možné pro Xamarin.Forms. Proto jsem se rozhodl pro alternativní přihlašování skenováním QR kódu, který si uživatel po přihlášení do webové aplikace vygeneruje.

Kapitola 5

Testování

Testování probíhalo především na reálném zařízení, protože emulátory různých mobilních telefonů nedokáží najít souřadnice GPS a tudíž by nebylo možné ověření funkčnosti mapy. Dalším problémem s emulátory je, že nedokáží zprostředkovat samotné focení, které je primární funkcí aplikace. Testy probíhaly primárně na mobilním telefonu Xiaomi Redmi 3, který má verzi Androidu 5.1.1 a displej velikosti 5.0 palců. Dále byla aplikace testována na mobilních zařízeních s nižší i vyšší verzí Androidu a na různých displejích.

Funkční testy probíhaly formou interview. Uživatelé byli seznámeni s funkcí aplikace, nikoli s ovládacími prvky a jejich použitím. Uživatel byl přiveden do oblasti, kde se nacházelo místo, kde byla historická fotografie pořízena. Následně si spustil aplikaci a snažil se bez pomoci vyfotit nový záběr. Při ovládání nahlas popisoval, jak se v aplikaci orientuje, na co se chystá klepnout, co se mu na aplikaci líbí a co by udělal jinak, jak by aplikaci vylepšil, nebo co by změnil. Poznámky jsem si zapisoval.

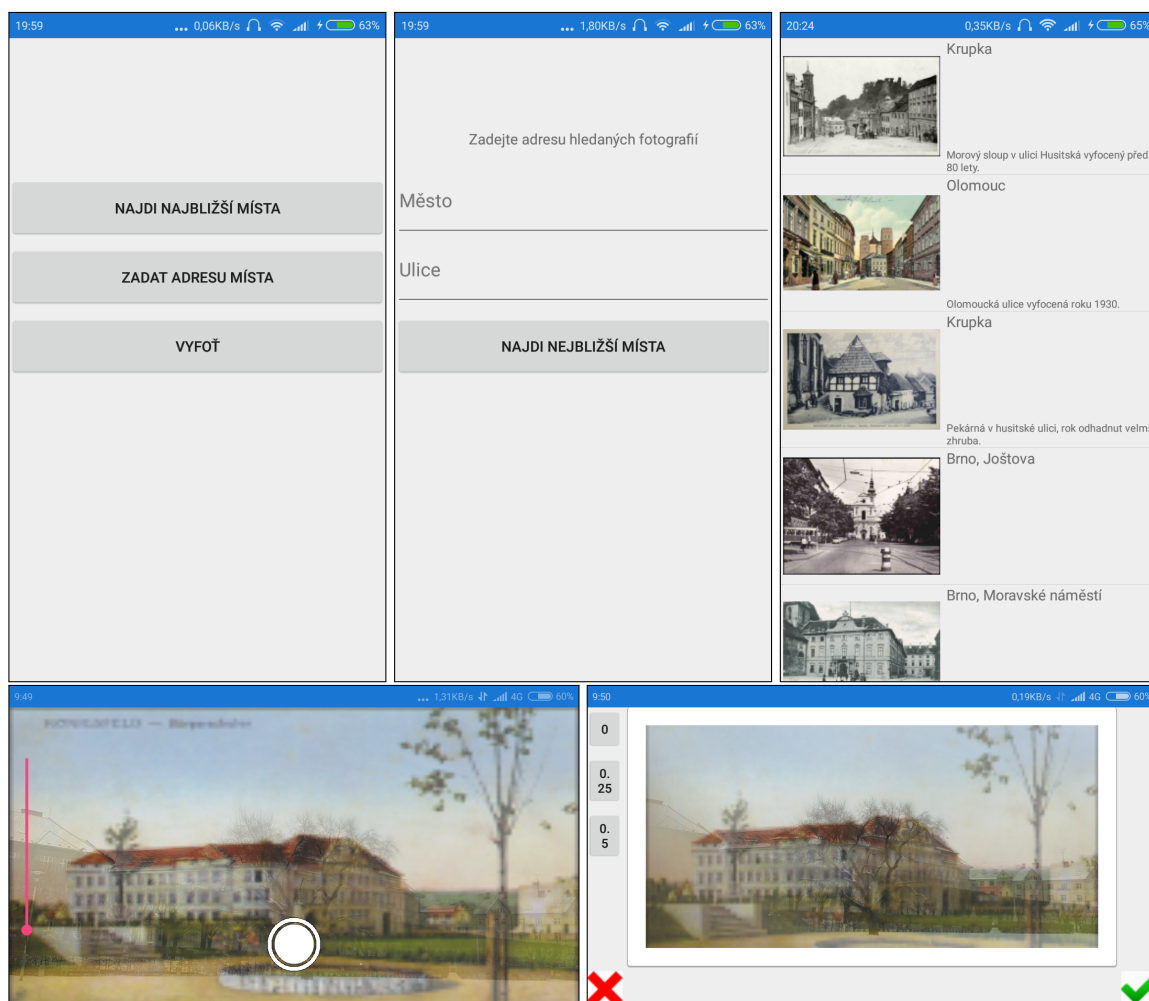
Při vyhodnocování jsem dal dohromady všechny poznámky a ty, které se objevily vícekrát, jsem v aplikaci pozměnil. Ostatní, které poznamenal pouze zlomek uživatelů, jsem ještě probral s technicky zdatnými uživateli. Jak už jsem se zmínil v kapitole 1, byla zajištěna databáze s více než 50 000 historických fotografií, avšak pro testování je použit jen zlomek z nich.

5.1 Aplikace verze 1.0

Po vydání první verze na začátku ledna 2017 byli vybráni 3 potenciální uživatelé, kteří ji testovali. Tento prototyp je zobrazen na obrázku 5.1. Nejvíce poznámek měli uživatelé k úvodní stránce, která vůbec nenaznačovala, k čemu aplikace je a co je její hlavní funkcí. Oblast okolo tří tlačítek byla prázdná a nebyla příliš vzhledná. Při vyhledávání místa podle adresy chybělo doplňování měst podle prvních zadaných písmen a opět bylo na stránce hodně prázdného místa. Na stránce s výpisem nebyli někteří uživatelé spokojeni s tím, že se musí vrátit na hlavní stránku, aby mohli fotografii vyfotit. U stránky, která umožňuje focení uživatelé neměli mnoho připomínek a nenapadalo je mnoho vylepšení, proto tato stránka do další verze zůstala nezměněná. Na stránce se srovnáním obou snímků uživatelé nerozuměli tlačítkům ke změně intenzity prolnutí a také nevěděli, co se stane po klepnutí na tlačítko potvrdit.

Ve výsledku mi dali uživatelé několik rad, které byly aplikovány do další verze programu:

- Úvodní stránka musí lépe přiblížit, k čemu aplikace slouží.
- Hlavní stránka je téměř zbytečná, bylo by dobré zpracovat nápad se zobrazením formou mapy.
- Tlačítka musí vypovídat o jejich funkci.
- Z výpisu se musí ihned přejít přímo k fotografování.



Obrázek 5.1: Na obrázku je vidět ukázka z aplikace ve verzi 1.0. Obrázek nahoře vlevo znázorňuje hlavní stránku, po klepnutí na „zadat adresu místa“ byl uživatel přesunut na stránku zobrazenou na obrázku nahoře uprostřed. Po vyhledání nejbližších míst nebo zadání adresy byla načtena stránka s výběrem místa (vpravo nahoře). Vlevo dole je obrázek se stránkou, která umožňuje focení a vpravo dole je znázorněna ukázka stránky, na které probíhalo srovnání.

5.2 Aplikace verze 1.1

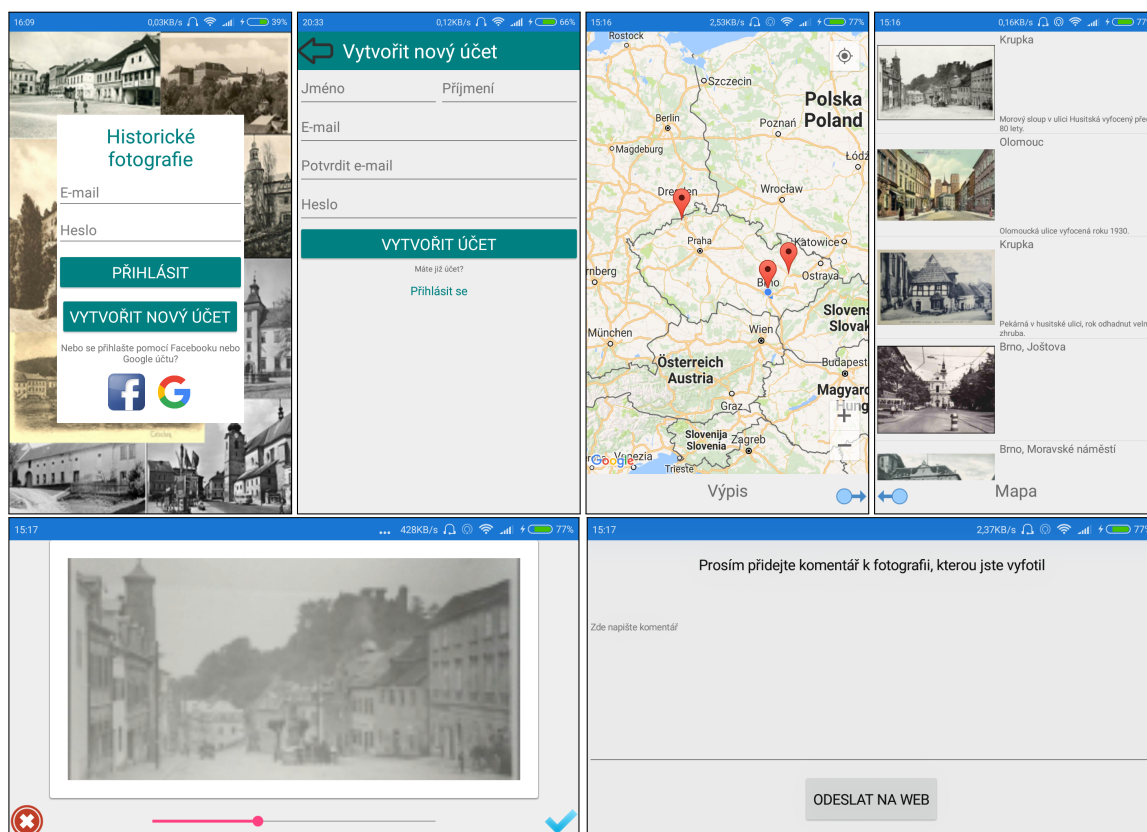
Do verze 1.1, která je znázorněná na obrázku 5.2, byly aplikovány poznámky, které uživatelé zdůraznili při testování první aplikace. Byla přidána stránka s přihlášením a vytvořením nového účtu, bylo navrženo přihlašování přes sociální sítě a vytvořena koláž, která vyplňuje pozadí přihlašovací stránky. Hlavní stránkou po přihlášení se stala mapa se zobrazenými body s fotografiemi a stránka výpisu. K tomu se dalo přecházet pomocí gesta tahu doprava nebo doleva v lokální navigaci. Na stránce srovnání byl přidán posuvník pro změnu intenzity prolnutí místo výše zmíněných tlačítek, po potvrzení srovnané fotografie se načetla stránka s přidáním komentáře. Na té se nachází tlačítko, které umožňuje odeslat pořízený snímek na webovou aplikaci.

Kromě lidí, kteří testovali první verzi aplikace, jsem požádal další dva, jež program ještě neviděli a nebyli ovlivněni srovnáváním s předchozí verzí, tudíž byly jejich poznatky nové. Vzhledem k tomu, že byla přidána mapa, bylo potřeba testovat i správnou funkčnost zobrazování fotografií podle GPS souřadnic a aktuální pozici zařízení. Při testování s uživateli byly opět zapisovány jejich poznatky. U přihlašovací stránky se objevily poznámky o názvu aplikace „Historické fotografie“, který opět nedokázal evokovat hlavní funkci. Na hlavní stránce s mapou nedokázala většina testerů rozpoznat, jak přejít na další stránku s výpisem zobrazených fotografií. Stejně tomu bylo i u stránky s výpisem. Na stránce se srovnáním chtěli mít uživatelé možnost náhledu pouze historické fotografie. Dalším objeveným problémem verze byla orientace v terénu, tedy jak nalézt, odkud byla fotografie pořízena. Výsledkem testování druhé verze programu byly tyto poznatky od uživatelů:

- Změnit logo aplikace.
- Vylepšit lokální navigaci na stránce s mapou a výpisem.
- Přidání náhledu historické fotografie pro lepší orientaci v terénu.

5.3 Finální verze aplikace

Do finální verze aplikace bylo přidáno alternativní přihlašování pomocí načtení QR kódu vygenerovaného webovou aplikací. Dále byla změněna lokální navigace na hlavní stránce s mapou a výpisem, která umožňuje snadněji přistoupit k oběma stránkám. Po výběru snímku, který chce uživatel vyfotit, se otevře nová stránka s náhledem historické fotografie. Na této stránce je možnost zobrazení poznámek, navigace k místu a vyfocení aktuálního snímku. Na stránce srovnání byl přidán náhled pouze historické fotografie pro přesnější zasazení do kontextu. Celá finální verze aplikace i s popisem jejího použití je popsána v sekci 4.4 nebo v příloze. Finální verze byla zveřejněna na oficiálním obchodě s aplikacemi Google Play.



Obrázek 5.2: Na obrázku je ukázka mobilní aplikace ve verzi 1.1. Nahoře vlevo je přihlašovací stránka, vedle ní napravo si uživatel zakládá nový účet. Stránka (druhá zleva nahoře) sloužila k zobrazování bodů s místy na mapě a vedle ní je zobrazena ukázka stránky s výpisem informací o místech. Vpravo dole je obrázek se stránkou srovnání fotografií a vedle ní napravo je stránka pro doplnění komentáře a následné odeslání fotografie na webovou aplikaci.

Kapitola 6

Závěr

Cílem praktické části mé bakalářské práce bylo vytvořit mobilní aplikaci pro pořizování nových záběrů historických pohlednic. Aplikací tohoto typu je nedostatek a věřím, že se mé řešení stane přínosem pro lidi zajímající se o historii. Aplikace je implementována pro mobilní operační systém Android. Aplikace nese název RePhoto a je volně ke stažení na Google Play.

Osobně mi aplikace přinesla mnoho zkušeností v programování. Seznámil jsem se s problematikou repeat photography a přístupem, jak vyvíjet aplikace multiplatformě. Velkým přínosem pro mě bylo pochopení fungování operačního systému Android a jeho SDK. Dále jsem si osvojil vývoj aplikací prototypováním. Programování v Xamarin a práce s NuGet balíčky pro mě byla mnohdy velice frustrující, ale rád jsem si tím prošel a získal mnoho zkušeností. Neméně znalostí jsem nabral v programovacím jazyce C[#].

Tato práce splnila své zadání, cílem teoretické části bylo čtenáře seznámit s problematikou repeat photography. Další věcí je představit čtenáři technologie, které byly v praktické části použity. Kapitola o návrhu aplikace lidem přiblíží vývoj softwaru metodou prototypování a seznámí ho se samotnou funkčností výsledného programu. Cílem části práce, která pojednává o implementaci, jsem se snažil čtenářům nastítnit praktiky a řešení problému v určitých situacích. Při testování došlo k odhalení mnoha chyb, které jsem postupně odstranil.

Aplikaci plánuji v budoucnu rošířit. Uživatelské rozhraní bylo psáno v multiplatformním kódu, tudíž první rozšíření je přetvořit aplikaci, aby byla kompatibilní i s ostatními mobilními operačními systémy. Dalším rozšířením je zvýšení množství historických fotografií v databázi. Aplikaci plánuji propojit se sociálními sítěmi a připojit možnost sdílet na nich porovnání fotografií. Při velkém počtu snímků by byla mapa zahlcena vyznačenými body, a proto plánuji navrhnout jejich propojování a zobrazení informací o fotografii již po klepnutí na bod. Plánuji snížit velikost instalačního souboru aplikace, který je příliš velký.

Literatura

- [1] H. WEBB, Robert, Diane E. BOYER a Raymond M. TURNER, ed. Repeat Photography: Methods and Applications in the Natural Sciences [online]. Connecticut Ave., NW, Washington, DC 20009: Island Press, 2010 [cit. 2017-04-27]. ISBN 978-1597267137. Dostupné z: <https://www.amazon.com/Repeat-Photography-Methods-Applications-Sciences/dp/1597267139>.
- [2] Daniel, S. F.: *Mastering Xamarin UI Development*. Packt Publishing, Jan 2017, ISBN 9781786462008.
- [3] Douglas Crockford: *JSONRequest*. Duben 2006, [Online; navštíveno 03.05.2017]. URL <http://json.org/JSONRequest.html>
- [4] Goodwin, D.; Coleman, B.: *Designing UX: Prototyping*. SitePoint, Mar 2017, ISBN 9780994347084.
- [5] Hermes, D.: *Xamarin Mobile Application Development: Cross-Platform C# and Xamarin.Forms Fundamentals*. Apress, 2015, ISBN 978-1-4842-0215-9.
- [6] PETR, I.: *Fotohistorie*. [Online; navštíveno 14.03.2017]. URL <http://fotohistorie.cz>
- [7] Petzold, C.: *Creating Mobile Apps with Xamarin.Forms*. Microsoft Press, 2016, ISBN 978-1-5093-0297-0.
- [8] Statista - The Statistics Portal: *Global mobile OS market share in sales to end users from 1st quarter 2009 to 1st quarter 2016*. [Online; navštíveno 20.04.2017]. URL <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems>

Přílohy

Příloha A

Obsah CD

- **report.pdf** – tato technická zpráva
- **latex/** – zdrojové soubory LaTeXu
- **src/** – zdrojové kódy aplikace
- **poster.pdf** – demonstrační plakát
- **video.mp4** – video demonstrující funkci aplikace
- **RePhoto.apk** – instalační soubor aplikace